Data Preperation Report

dataMineR

19/6/2013

Contents

CHAPTER 1

INTRODUCTION

This analysis report is generated using R, R-studio and knitr to knitr R code from markdown into html and later LateX format. We have the option to include all R code that is used to generate the plots and calculations. By default this feauture is dissabled.

The data preperation step is the second step in a datamining analysis. Steps identified in the datamining process are:

- Data analysis
- Behaviour analysis
- Missing value analysis
- Missing value imputation (optional)
- Binning
- Feature selection
- Model development
- Model analysis
- Model deployment

1.1 Information on Dataset

Basic information from the dataset we are using.

We are using data from file : ../data/data-set.Rdata. The dataset has 26 variables and 92467 rows.

1.2 Variabele types

The following variabeles are present in the dataset: age, customer_lifetime, contact_channel, product_grp, action, sales_channel, months_under_contract, P1_amount, contract_type, usage, usage2, donwpayment, extra_payment, city_code, unkown_amount, product_detail, tax_amount, P2-amount, P1_turnover, P2_turnover, index1, days_since_last_sale_p1, days_since_last_sale_p2, days_since_last_contract_adj, days_since_last_sale, target

We have 19 numeric variables and 7 categorical variables (or factors in R).

1.3 Target defenition

This analysis aims to report of the behaviour of each individual 'predictor' to a target variable. The target variable should be a categorical variable having two categories(or factor levels).

The target variable is defined in the previous step.

The target has the following proportion of outcomes:

```
## Attaching package: 'pander'
```

```
## The following object is masked from 'package:knitr':
##
## pandoc
```

	count	%
N	76680	82.93
Y	15787	17.07

1.4 Missing values

Before we can do model analysis we need to take care of missing values. The simplest appraoch,if the missing data in Missing Completly at Random(MCAR) or Missing at Random(MAR), is to delete cases including one or more missing entries but this can remove a large proportion of valuable data.

We can also remove individual variables if they have a high percentage of missing atributes. Or we can replace or impute missing data with for instance an average or most frequent value. In R we can use the function na.roughfix() from the package randomForest for this. Actually changing the data. For this we can use kNN nearest neighbors algorithm implemented in package DMwR or rfImpute() from the package RandomForest.

For now we will impute using na.roughfix().

This dataset has 0 incomplete cases out of 92467, which is 0 percent.

```
## Loading required package: randomForest
```

```
## randomForest 4.6-7
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

1.5 Variable interactions

Hierarchical clustering is a good technique to visualize interactions between numeric variables.

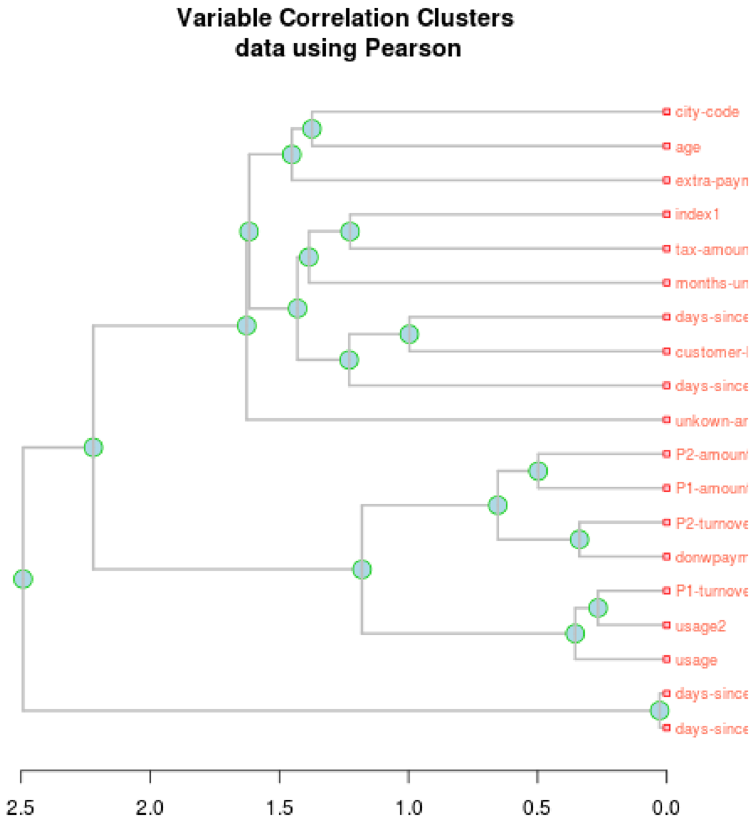


Figure 1.1: plot of chunk cluster

Chapter 2

Behaviour analysis summary

The selected inputs have the following raw predictive capacity for predicting the target outcome:

```
## Loading required package: survival
```

```
## Loading required package: splines
```

```
## Loading required package: Formula
```

```
## Hmisc library by Frank E Harrell Jr
##
## Type library(help='Hmisc'), ?Overview, or ?Hmisc.Overview') to see overall
## documentation.
```

```
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:survival':
##
## untangle.specials
```

```
## The following object is masked from 'package:randomForest':
##
## combine
```

```
## The following object is masked from 'package:base':
##
## format.pval, round.POSIXt, trunc.POSIXt, units
```

variable	Kendalls Tau	Somers Dxy
product_detail	-0.1156	-0.2117
months_under_contract	-0.09281	-0.1672
age	-0.04564	-0.08661
unkown_amount	-0.02596	-0.04905
city_code	-0.006576	-0.01238

contact_channel	0.01124	0.0203
extra_payment	0.01756	0.02863
sales_channel	0.01966	0.03517
usage2	0.0246	0.04668
usage	0.02715	0.05152
days_since_last_sale_p1	0.02854	0.05073
days_since_last_sale_p2	0.02856	0.05124
P1_amount	0.02966	0.05615
product_grp	0.03664	0.06479
contract_type	0.04086	0.06435
donwpayment	0.04273	0.08074
customer_lifetime	0.04742	0.08827
P1_turnover	0.04915	0.09324
action	0.06138	0.1111
P2-amount	0.06357	0.1203
P2_turnover	0.07075	0.1342
index1	0.09422	0.1788
days_since_last_sale	0.104	0.1947
days_since_last_contract_adj	0.1084	0.2051
tax_amount	0.1167	0.2046

Table 2.1: Rank correlation measures!

CHAPTER 3

Detailed variable behaviour analysis

Now we look at individual variables and analyse it's relation with the target. We visualize numerical variables different then categorical ones.

3.1 Behaviour for variable age

Here we wil look into the relation of variable age with the target.

```
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:Hmisc':
##
## is.discrete, summarize
```

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

3.2 Behaviour for variable customer_lifetime

Here we wil look into the relation of variable customer_lifetime with the target.

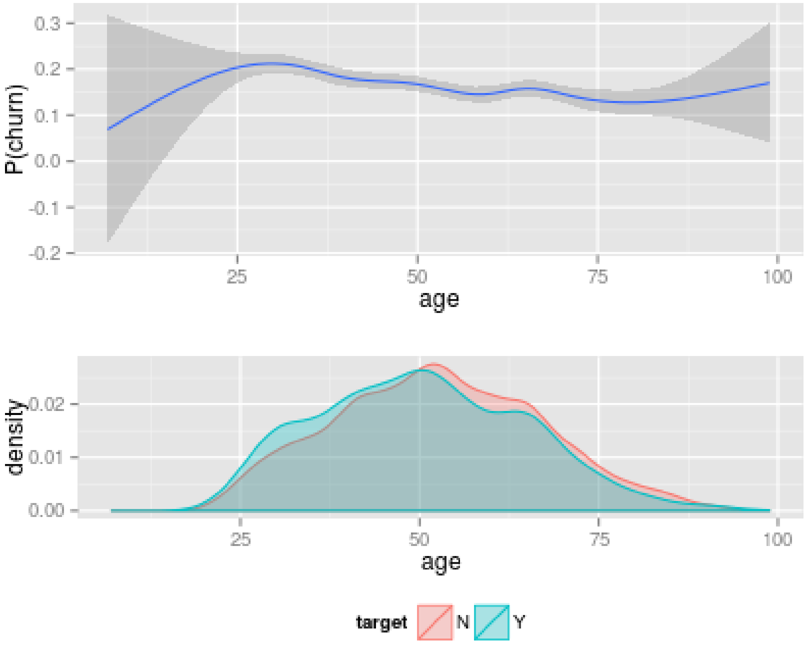
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

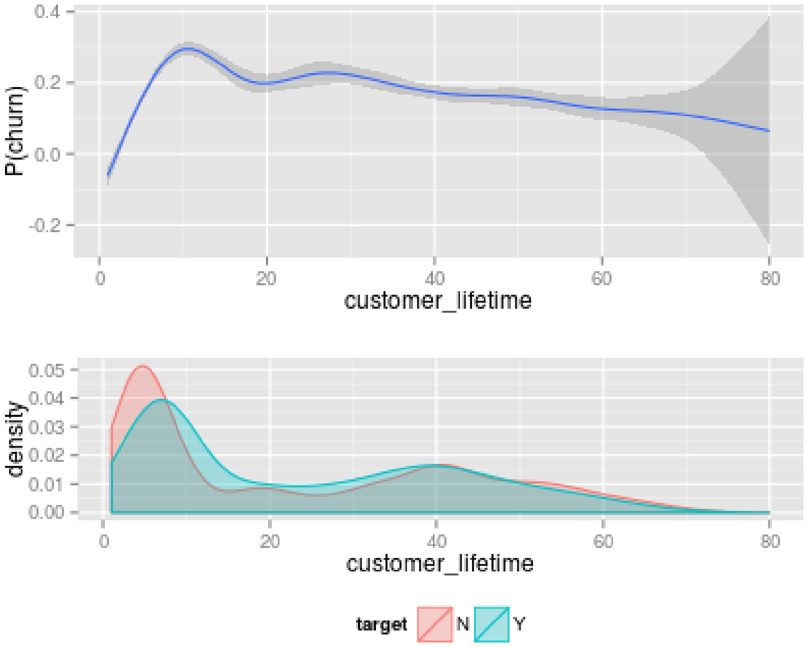
3.3 Behaviour for variable contact_channel

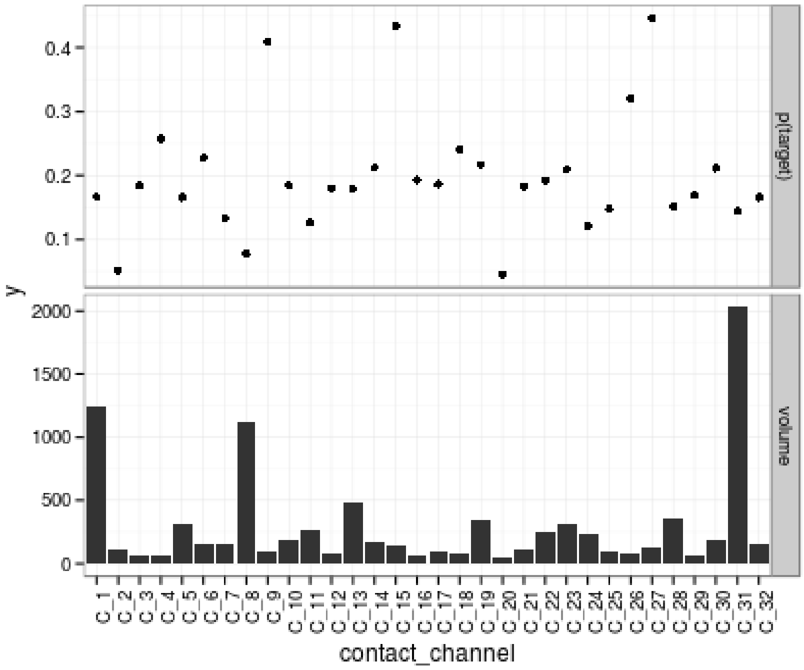
Here we wil look into the relation of variable contact_channel with the target.

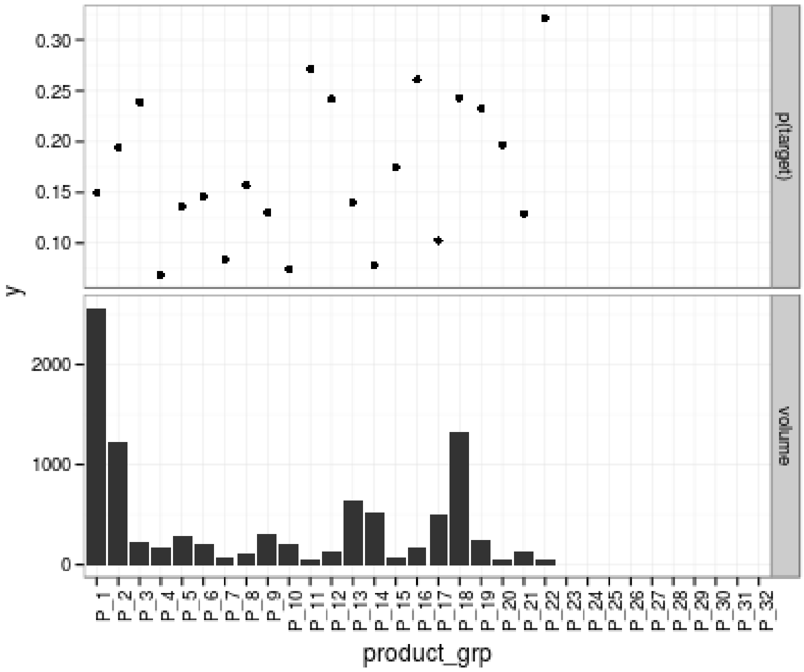
3.4 Behaviour for variable product_grp

Here we wil look into the relation of variable product_grp with the target.


P(churn)
0.3
0.2
0.1
0.0
-0.1
-0.2
25
50
75
100
age
density
0.02
0.01
0.00
25
50
75
100
age
target
N
Y

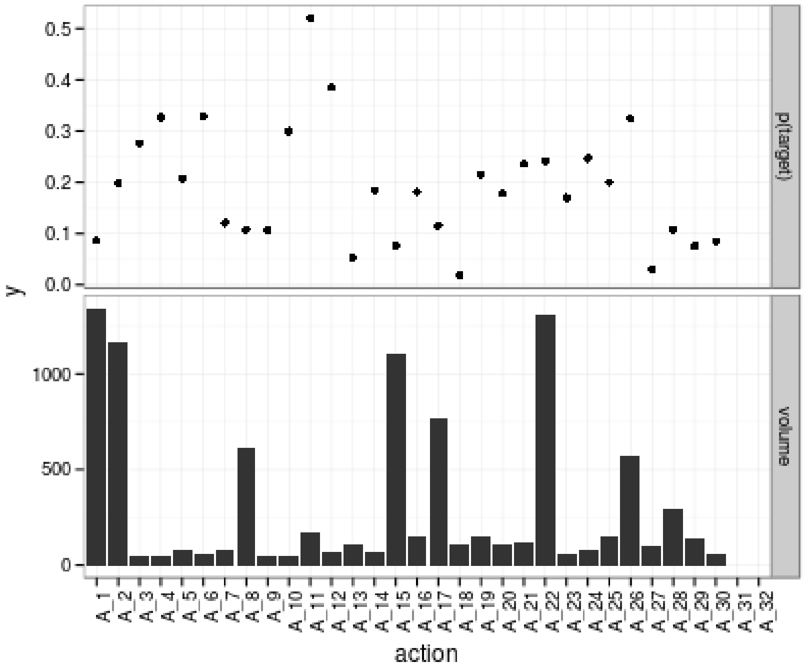

P(churn)
0.4
0.2
0.0
-0.2
0
20
40
60
80
customer_lifetime
density
0.05
0.04
0.03
0.02
0.01
0.00
0
20
40
60
80
customer_lifetime
target
N
Y


y
0.1
0.2
0.3
0.4
0
500
1000
1500
2000
p(target)
volume
C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_10 C_11 C_12 C_13 C_14 C_15 C_16 C_17 C_18 C_19 C_20 C_21 C_22 C_23 C_24 C_25 C_26 C_27 C_28 C_29 C_30 C_31 C_32
contact_channel


y
0.10
0.15
0.20
0.25
0.30
0
1000
2000
p(target)
volume
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_10 P_11 P_12 P_13 P_14 P_15 P_16 P_17 P_18 P_19 P_20 P_21 P_22 P_23 P_24 P_25 P_26 P_27 P_28 P_29 P_30 P_31 P_32
product_grp

3.5 Behaviour for variable action

Here we wil look into the relation of variable action with the target.



3.6 Behaviour for variable sales_channel

Here we wil look into the relation of variable sales_channel with the target.

3.7 Behaviour for variable months_under_contract

Here we wil look into the relation of variable months_under_contract with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

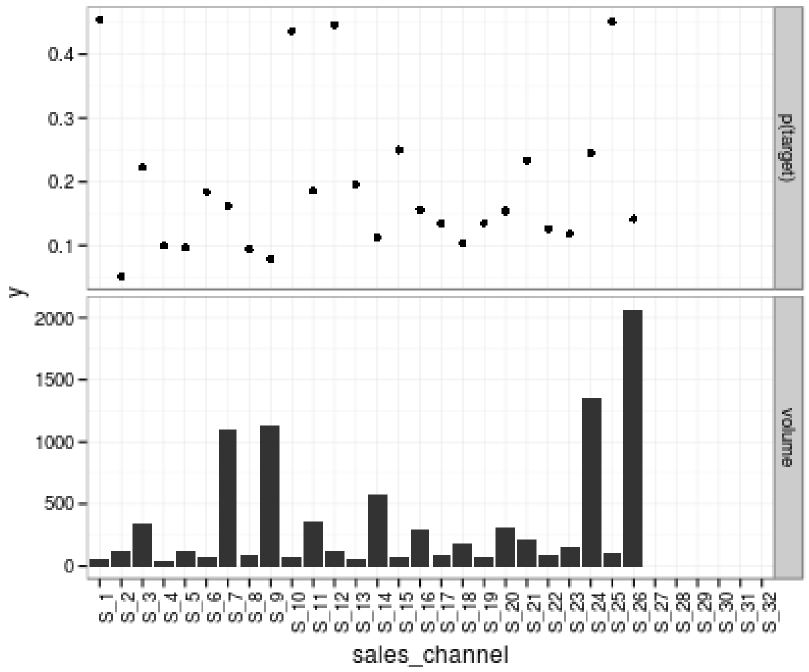
3.8 Behaviour for variable P1_amount

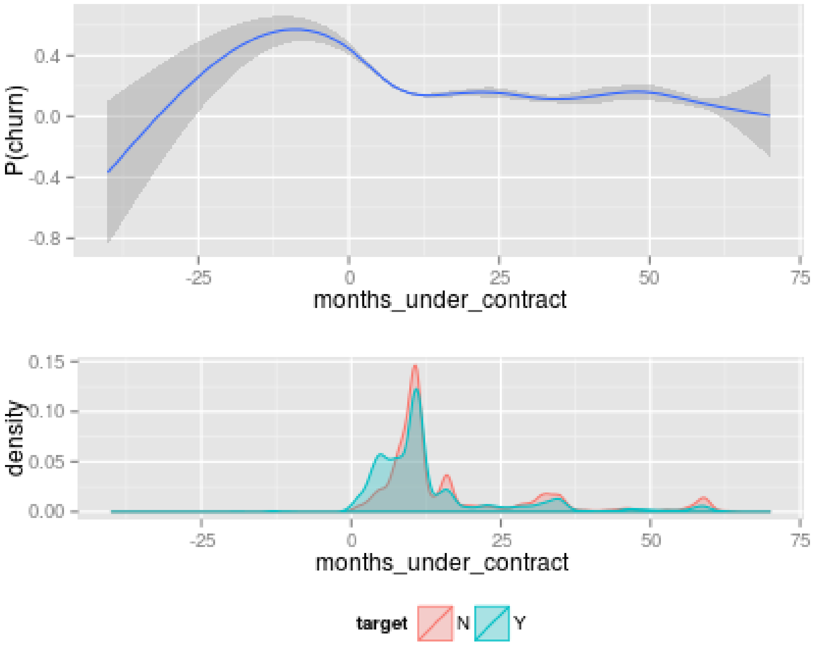
Here we wil look into the relation of variable P1_amount with the target.

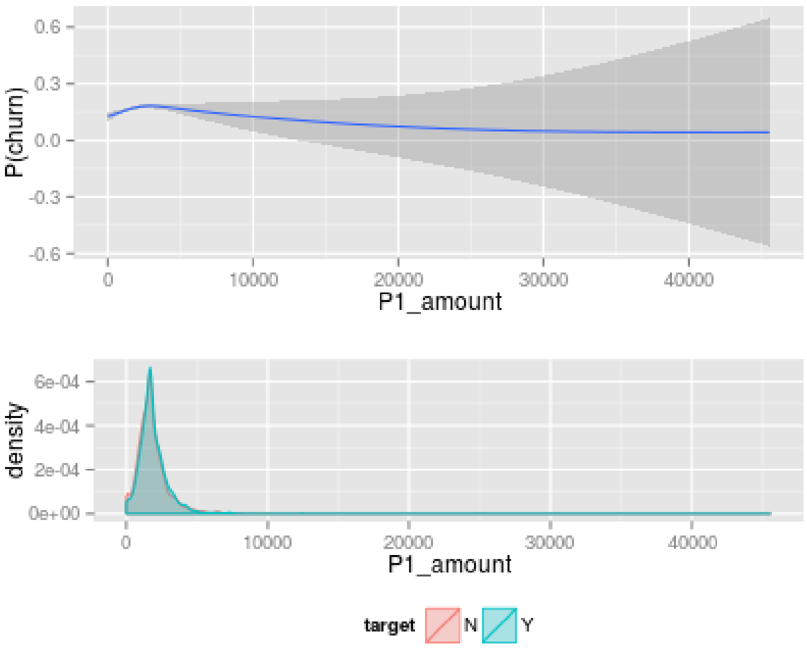
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

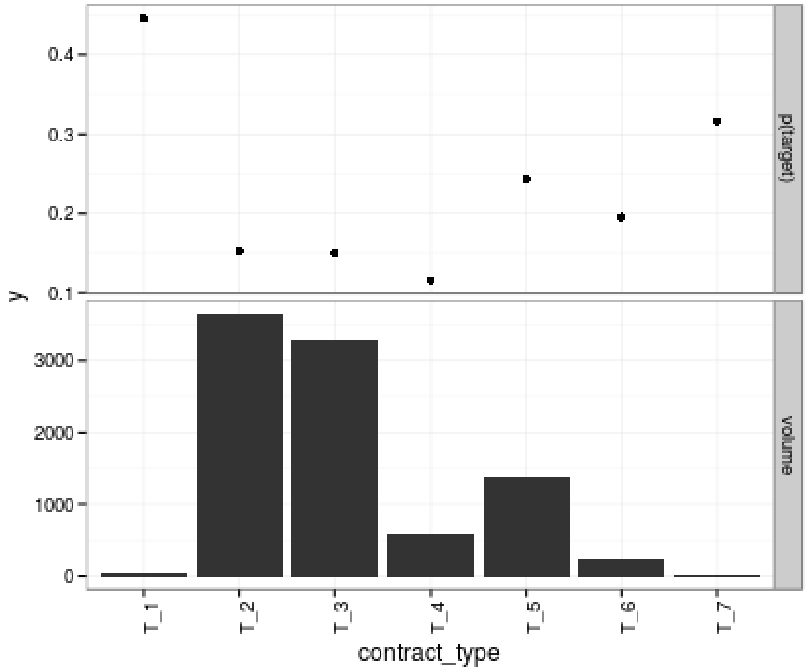
3.9 Behaviour for variable contract_type

Here we wil look into the relation of variable contract_type with the target.


0.4
0.3
0.2
0.1
y
2000
1500
1000
500
0
p(target)
volume
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
S_9
S_10
S_11
S_12
S_13
S_14
S_15
S_16
S_17
S_18
S_19
S_20
S_21
S_22
S_23
S_24
S_25
S_26
S_27
S_28
S_29
S_30
S_31
S_32
sales_channel


0.4
0.0
-0.4
-0.8
P(churn)
-25
0
25
50
75
months_under_contract
0.15
0.10
0.05
0.00
density
-25
0
25
50
75
months_under_contract
target
N
Y

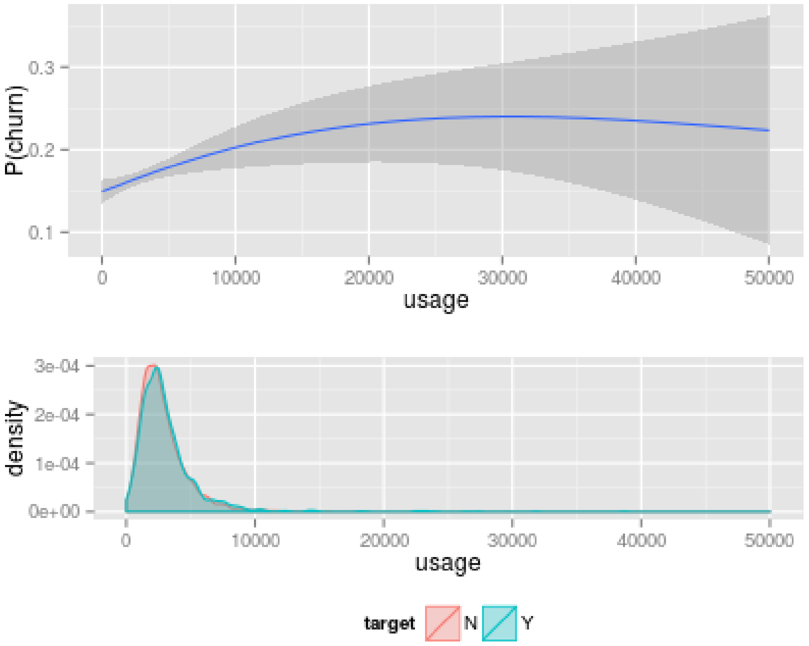

P(churn)
0.6
0.3
0.0
-0.3
-0.6
0
10000
20000
30000
40000
P1_amount
density
6e-04
4e-04
2e-04
0e+00
0
10000
20000
30000
40000
P1_amount
target
N
Y


y
0.4
0.3
0.2
0.1
p(target)
3000
2000
1000
0
volume
T_1
T_2
T_3
T_4
T_5
T_6
T_7
contract_type

3.10 Behaviour for variable usage

Here we wil look into the relation of variable usage with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



3.11 Behaviour for variable usage2

Here we wil look into the relation of variable usage2 with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

3.12 Behaviour for variable donwpayment

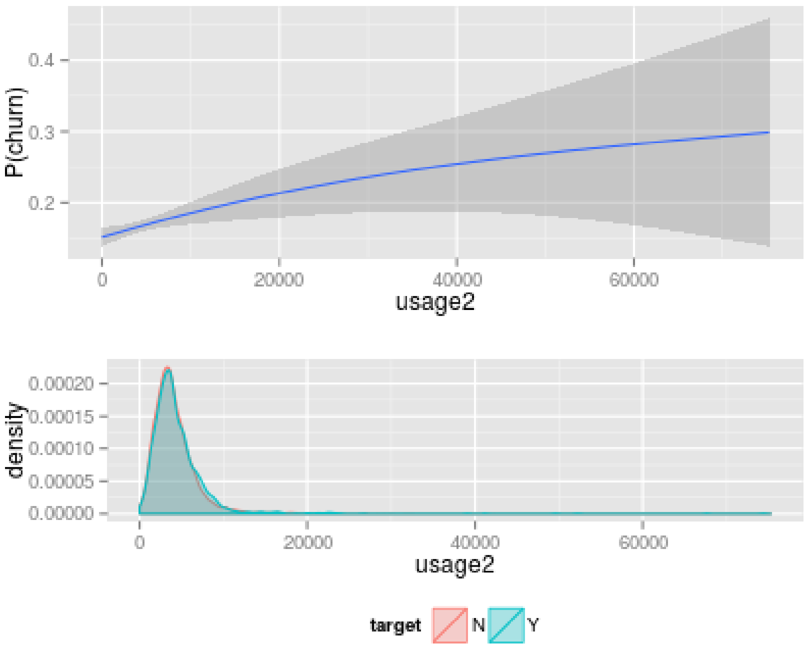
Here we wil look into the relation of variable donwpayment with the target.

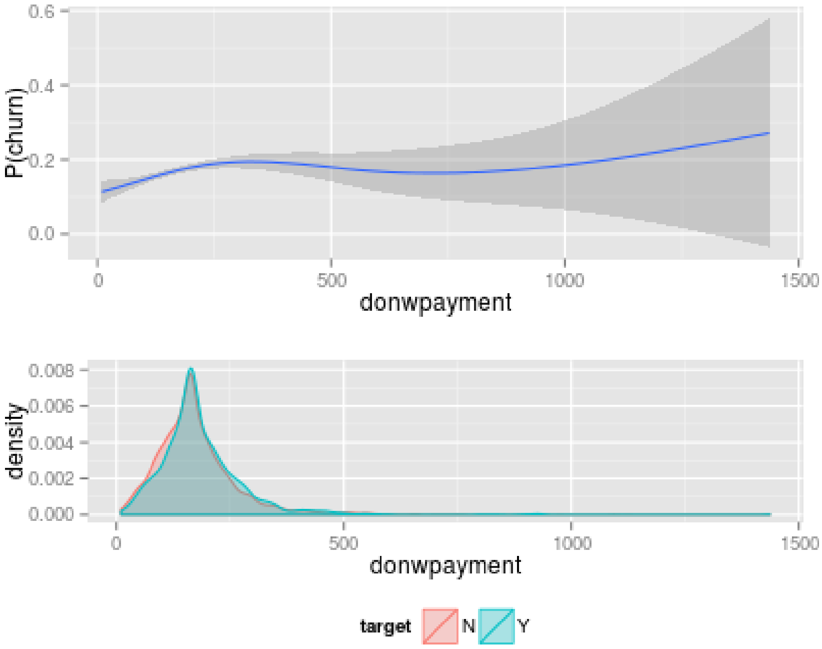
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

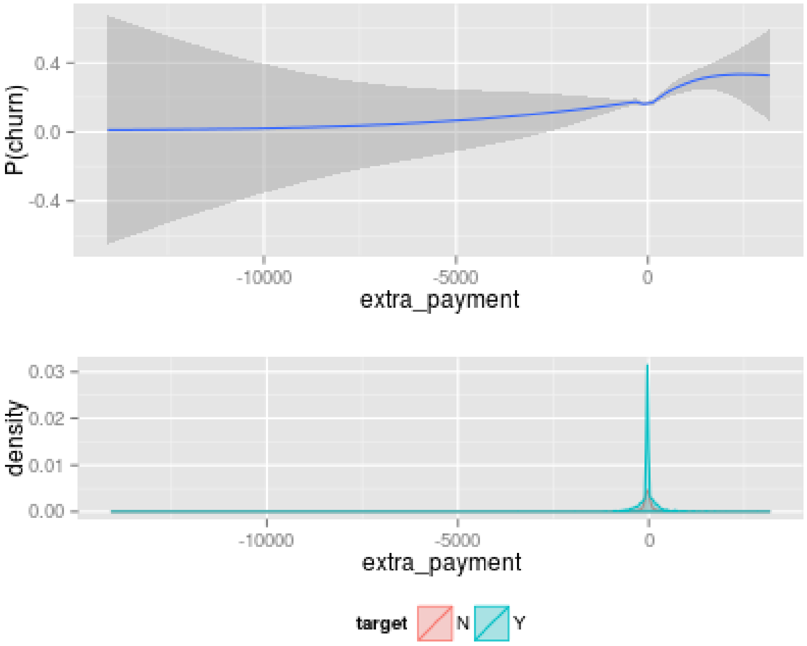
3.13 Behaviour for variable extra_payment

Here we wil look into the relation of variable extra_payment with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```


P(churn)
0.2
0.3
0.4
0
20000
40000
60000
usage2
density
0.00000
0.00005
0.00010
0.00015
0.00020
0
20000
40000
60000
usage2
target
N
Y

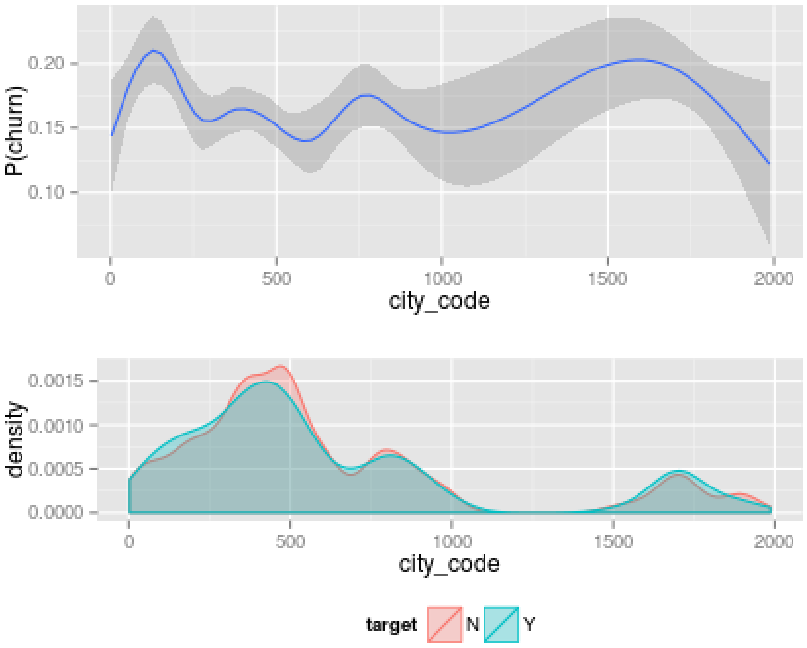

P(churn)
0.0
0.2
0.4
0.6
0
500
1000
1500
donwpayment
density
0.000
0.002
0.004
0.006
0.008
0
500
1000
1500
donwpayment
target
N
Y



3.14 Behaviour for variable city_code

Here we wil look into the relation of variable city_code with the target.

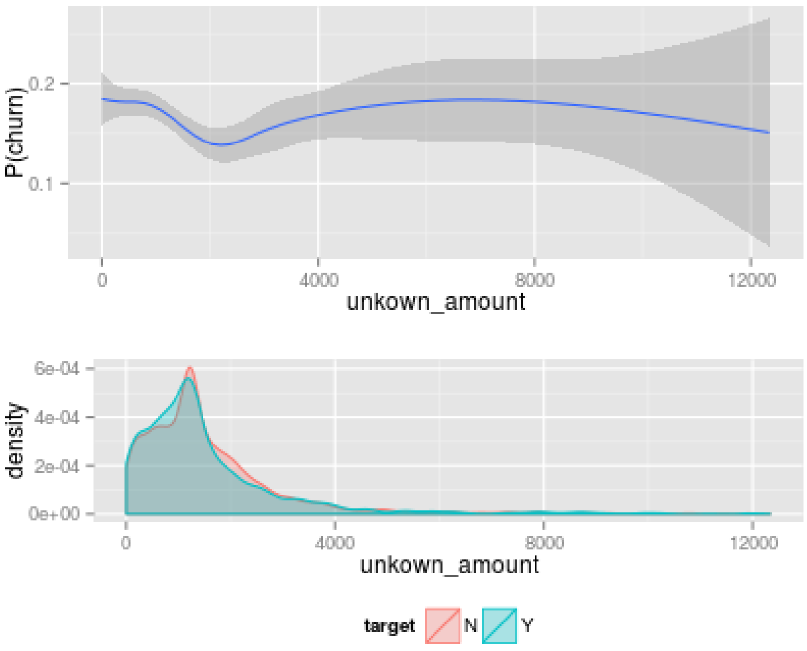
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



3.15 Behaviour for variable unkown_amount

Here we wil look into the relation of variable unkown_amount with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



3.16 Behaviour for variable product_detail

Here we wil look into the relation of variable product_detail with the target.

3.17 Behaviour for variable tax_amount

Here we wil look into the relation of variable tax_amount with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

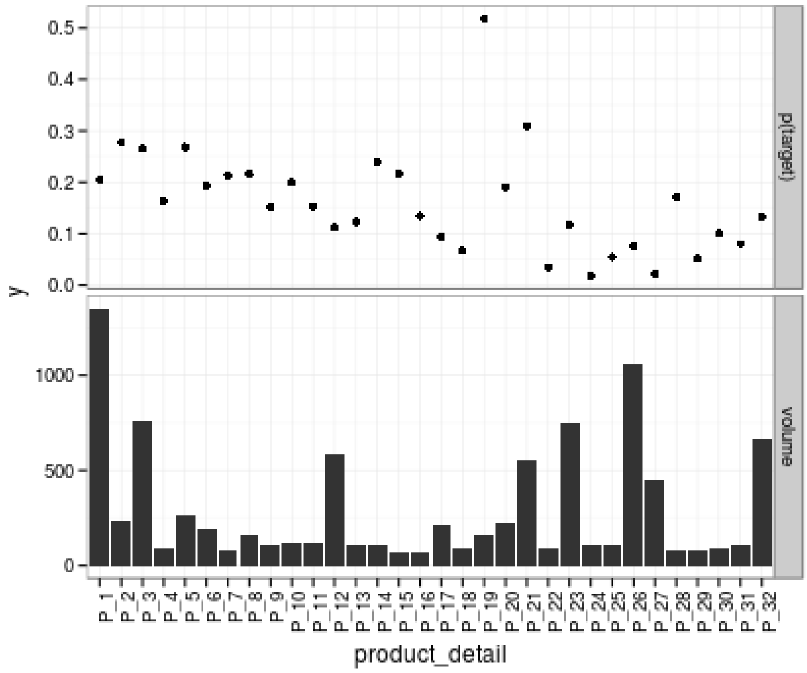
3.18 Behaviour for variable P2-amount

Here we wil look into the relation of variable P2-amount with the target.

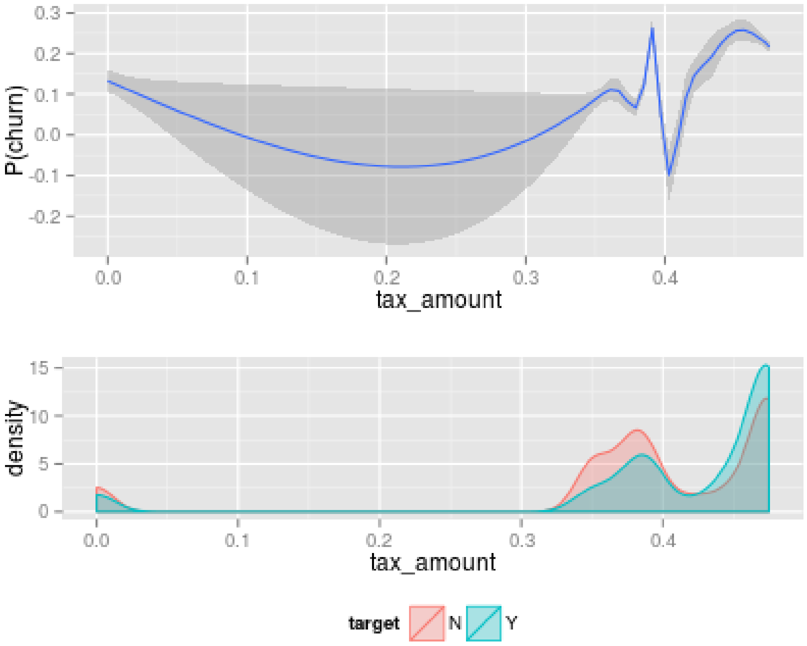
```
## Error: object 'P2' not found
```

3.19 Behaviour for variable P1_turnover

Here we wil look into the relation of variable P1_turnover with the target.

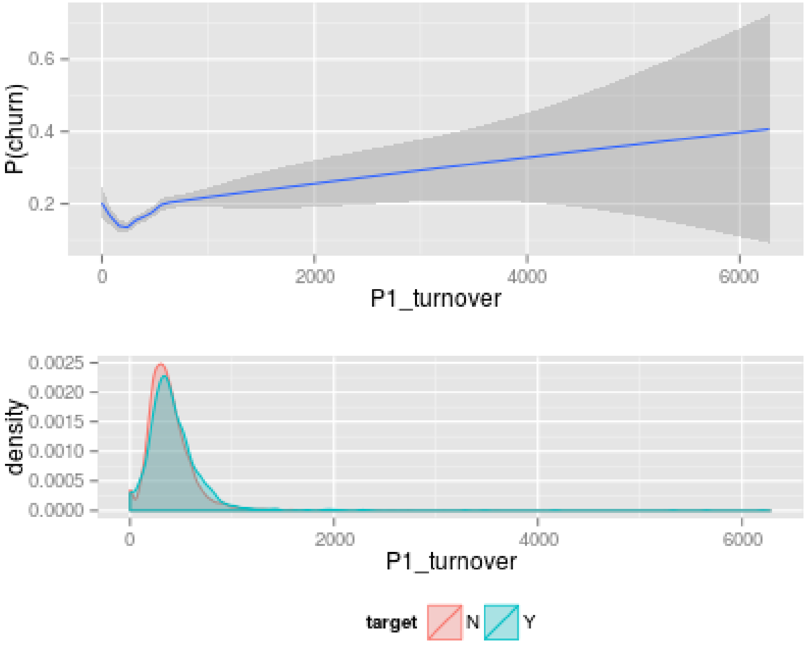


y
0.5
0.4
0.3
0.2
0.1
0.0
p(target)
1000
500
0
volume
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_10
P_11
P_12
P_13
P_14
P_15
P_16
P_17
P_18
P_19
P_20
P_21
P_22
P_23
P_24
P_25
P_26
P_27
P_28
P_29
P_30
P_31
P_32
product_detail



P(churn)
0.3
0.2
0.1
0.0
-0.1
-0.2
0.0
0.1
0.2
0.3
0.4
tax_amount
density
15
10
5
0
0.0
0.1
0.2
0.3
0.4
tax_amount
target
N
Y

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



3.20 Behaviour for variable P2_turnover

Here we wil look into the relation of variable P2_turnover with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

3.21 Behaviour for variable index1

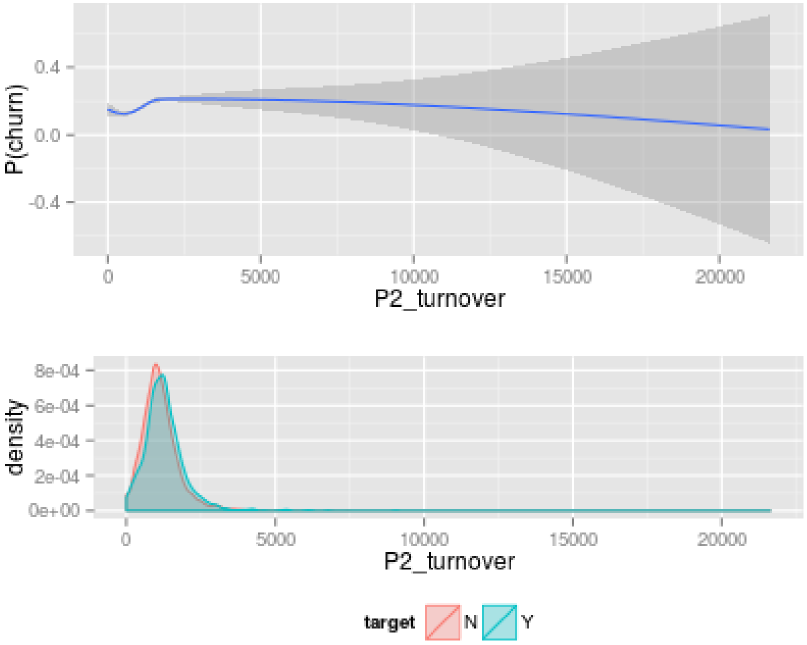
Here we wil look into the relation of variable index1 with the target.

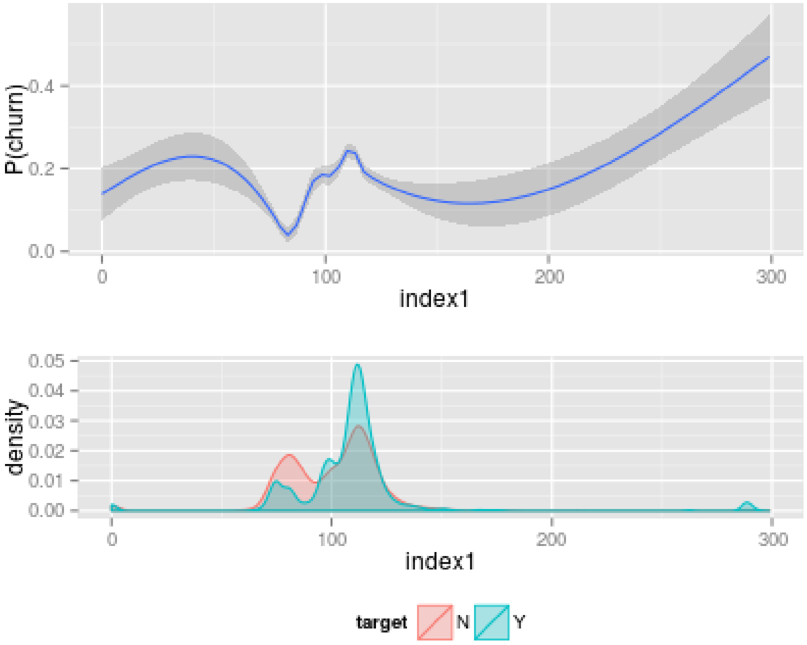
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

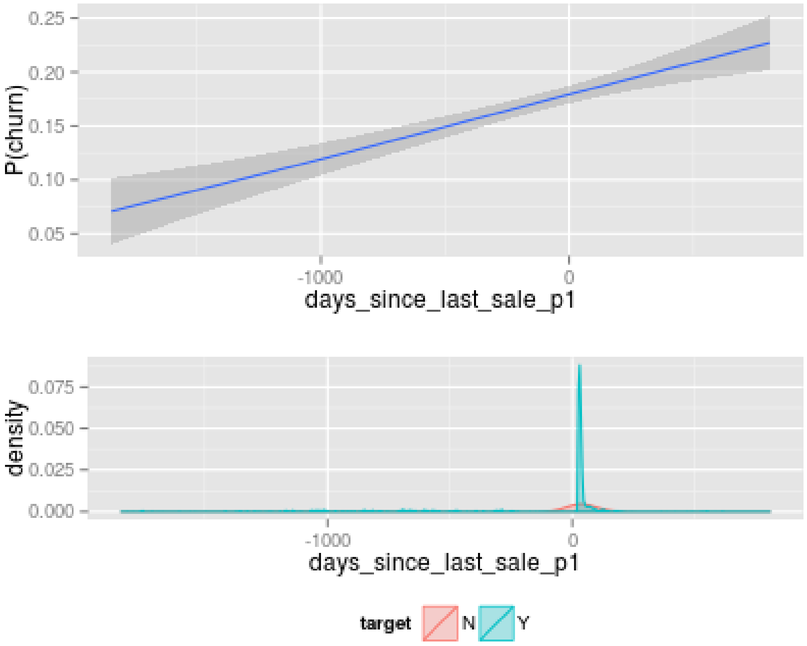
3.22 Behaviour for variable days_since_last_sale_p1

Here we wil look into the relation of variable days_since_last_sale_p1 with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```


P(churn)
0.4
0.0
-0.4
0
5000
10000
15000
20000
P2_turnover
density
8e-04
6e-04
4e-04
2e-04
0e+00
0
5000
10000
15000
20000
P2_turnover
target
N
Y

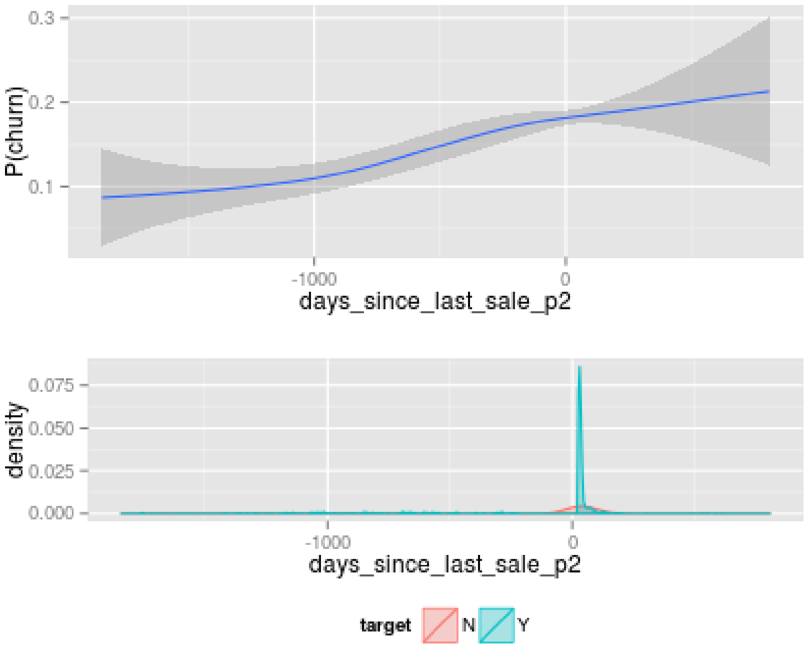

P(churn)
0.4
0.2
0.0
0
100
200
300
index1
density
0.05
0.04
0.03
0.02
0.01
0.00
0
100
200
300
index1
target
N
Y



3.23 Behaviour for variable days_since_last_sale_p2

Here we wil look into the relation of variable days_since_last_sale_p2 with the target.

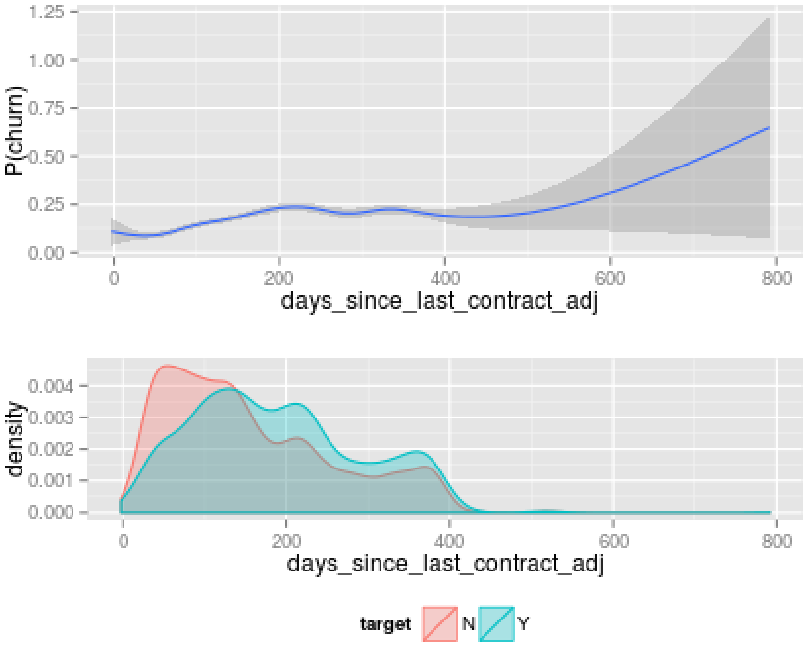
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



3.24 Behaviour for variable days_since_last_contract_adj

Here we wil look into the relation of variable days_since_last_contract_adj with the target.

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```



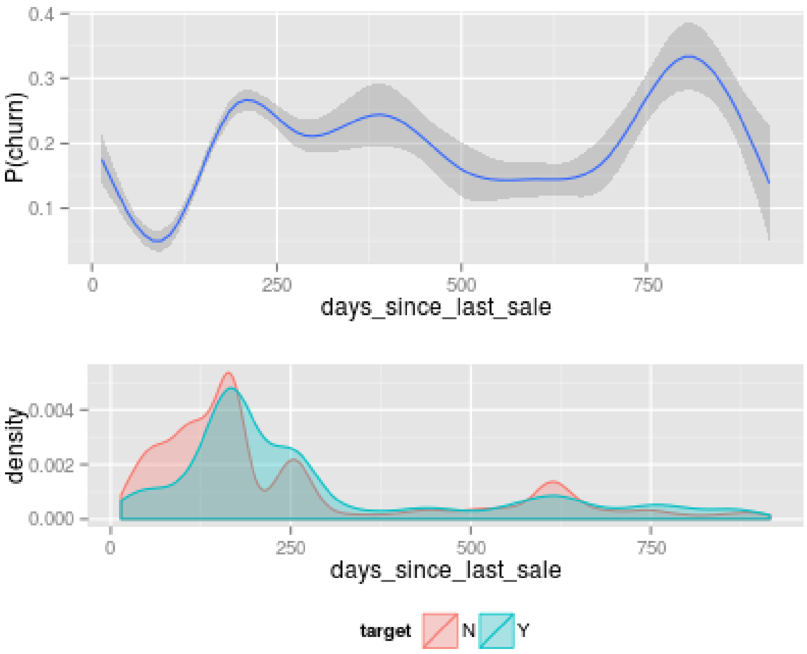
3.25 Behaviour for variable days_since_last_sale

Here we wil look into the relation of variable days_since_last_sale with the target.

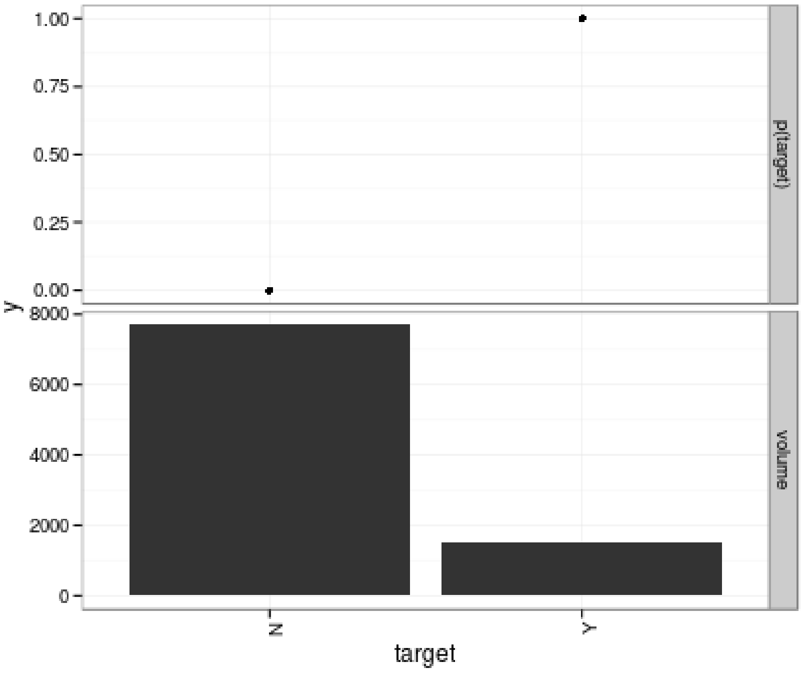
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
## gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
## smoothing method.
```

3.26 Behaviour for variable target

Here we wil look into the relation of variable target with the target.



P(churn)
0.4
0.3
0.2
0.1
0
250
500
750
days_since_last_sale
density
0.004
0.002
0.000
0
250
500
750
days_since_last_sale
target
N
Y



y
1.00
0.75
0.50
0.25
0.00
p(target)
8000
6000
4000
2000
0
volume
N
Y
target

CHAPTER 4

MODEL SET

We save the resulting set for moddeling as ‘r datasetName’.

The next step in Model development.